IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application:	:	Group Art Unit: 2446
Nicholas James Midgley	:	Examiner: Shaq Taha
Serial No.: 10/526,810	:	IBM Corporation
Filed: 03/04/2005	:	Intellectual Property Law
Title: REMOTE DYNAMIC CONFIGURATION	:	Department SHCB/040-3
OF A WEB SERVER TO FACILITATE	:	1701 North Street
CAPACITY ON DEMAND	:	Endicott, NY 13760
Confirmation No.: 8811		

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

I. Real Party in Interest

International Business Machines Corporation is the real party in interest.

II. Related Appeals and Interferences

There are no related Appeals or Interferences.

III. Status of Claims

Claims 1-62 were previously Canceled.

Claims 63-77 are Pending Finally Rejected and Appealed.

IV. Status of Amendments

A Response was filed on February 2, 2010, After Final Rejection, presenting Remarks to distinguish a newly cited reference, but **no** claim amendments. This did not result in allowance of the patent application.

V. Summary of Claimed Subject Matter

Support for claim elements is indicated in plain brackets [ ].

Claim 63 recites a method for allocating an additional real application server to an existing pool of real application servers. [Management Server 140 with Resource Allocation module 235, Analyser module 230 and Resource Update module 240. Application servers 145, 150 and 155, with Application/Dynamic content module 215, of Figure 1 are in a cluster. Page 8 line 38 to Page 9 line 27. As indicated therein, the additional application server is similar to server 145 which includes Application/Dynamic content module 215.] The pool includes a first real application server having an application installed therein and communicating with a real data-source server to obtain application data from the data-source server. [Application servers 145, 150 and 155 of Figure 1 are in a cluster. Page 7 lines 11-25. Application/Dynamic content module 215 within application servers 145, 150 and 155. Page 7 lines 26- 42. Data source server 175 or 180. Page 5 lines 36-37. Page 8 lines 10-15. Figure 1 illustrates a network path including an optional router or firewall 165 and optional additional network 170 which interconnects application servers 145, 150 and 155 to data-source servers 175 and 180. Page 5 line 40 to Page 6 line 2.] The additional application server has the application installed therein. [Page 8 line 38 to Page 9 line 27. As indicated therein, the additional server is similar to server 145 which includes Application/Dynamic content module 215.] A real management server for the pool receives performance data for the first real application server and performance data for the real data-source server. [Management server 140 of Figure 2. Page 8 lines 16-36. Analyser module 230 collects performance data. Steps 500, 505 and 510. Reporting module 205 of

Figure 2 or Reporting module 210 of Figure 3. Step 355 of Figure 3. Steps 400, 405 and 410 of Figure 4. Page 7 line 42 to Page 8 line 5. Page 8 lines 16-26. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] The real management server, based on the performance data for the first real application server and the performance data for the real data-source server, automatically determining that the first real application server is functional but has reached a predetermined upper level of utilization. [Analyser Module 230 of Figures 2 and 5. Page 8 lines 28-36. Steps 500, 505 and 510.] The performance data for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. [Page 7 line 42 to Page 8 line 5. Page 8 lines 16-26. Steps 500, 505 and 510. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] In response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization, the real management server automatically identifies the additional real application server as having the application but not currently allocated to the pool, [Resource Allocation module 235 in server 140. Page 8 line 38 to Page 9 line 15. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] and the real management server automatically selects the real data-source server to provide application data to the additional real application server and automatically sends connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server. [Resource Allocation module 235 in server 140. Dynamic Configuration module 216. Page 8 line 38 to Page 9 line 27. Figure 3 illustrates "Connection Configuration" file 320. Page 8 lines 7-9. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 20.]

Claim 68 recites a real management server for allocating an additional real application server to an existing pool of real application servers. [Management Server 140 with Resource Allocation module 235, Analyser module 230 and Resource Update module 240. Application servers 145, 150 and 155, with Application/Dynamic content module 215, of Figure 1 are in a cluster. Page 8 line 38 to Page 9 line 27. As indicated therein, the additional application server is similar to server 145 which includes Application/Dynamic content module 215.] The pool

includes a first real application server having an application installed therein and communicating with a real data-source server to obtain application data from the data-source server. [Application servers 145, 150 and 155 of Figure 1 are in a cluster. Page 7 lines 11-25. Application/Dynamic content module 215 within application servers 145, 150 and 155. Page 7 lines 26- 42. Data source server 175 or 180. Page 5 lines 36-37. Page 8 lines 10-15. Figure 1 illustrates a network path including an optional router or firewall 165 and optional additional network 170 which interconnects application servers 145, 150 and 155 to data-source servers 175 and 180. Page 5 line 40 to Page 6 line 2.] The additional application server has the application installed therein. [Page 8 line 38 to Page 9 line 27. As indicated therein, the additional server is similar to server 145 which includes Application/Dynamic content module 215.] The real management server includes a CPU, a computer readable memory and a computer readable storage media. [CPU 231, RAM 233 and storage 234 added to Figure 2 by Amendment of 8/29/08 and corresponding addition to Specification on Page 5 line 26 by Amendment of 8/29/08.] First program instructions receive performance data for the first real application server and performance data for the real data-source server. [Management server 140 of Figure 2. Page 8 lines 16-36. Analyser module 230 collects performance data. Steps 500, 505 and 510. Reporting module 205 of Figure 2 or Reporting module 210 of Figure 3. Step 355 of Figure 3. Steps 400, 405 and 410 of Figure 4. Page 7 line 42 to Page 8 line 5. Page 8 lines 16-26. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] Second program instructions, based on the performance data for the first real application server and the performance data for the real data-source server, determine that the first real application server is functional but has reached a predetermined upper level of utilization. [Analyser Module 230 of Figures 2 and 5. Page 8 lines 28-36. Steps 500, 510 and 515.] The performance data for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. [Page 7 line 42 to Page 8 line 5. Page 8 lines 16-26. Steps 500, 505 and 510. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] In response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization, the second program instructions identify the additional real application server as having the application but not currently allocated to the pool, [Resource Allocation module 235

in server 140. Page 8 line 38 to Page 9 line 15. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] and the second program instructions select the real data-source server to provide application data to the additional real application server and automatically send connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server. [Resource Allocation module 235 in server 140. Dynamic Configuration module 216. Page 8 line 38 to Page 9 line 27. Figure 3 illustrates "Connection Configuration" file 320. Page 8 lines 7-9. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 20.] The first and second program instructions are stored on the computer readable storage media for execution by the CPU via the computer readable memory. [CPU 231, RAM 233 and storage 234 added to Figure 2 by Amendment of 8/29/08 and corresponding addition to Specification on Page 5 line 26 by Amendment of 8/29/08. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 20.]

Claim 73 recites a computer program product for execution in a real management server for allocating an additional real application server to an existing pool of real application servers. [Management Server 140 with Resource Allocation module 235, Analyser module 230 and Resource Update module 240. Application servers 145, 150 and 155, with Application/Dynamic content module 215, of Figure 1 are in a cluster. Page 8 line 38 to Page 9 line 27. As indicated therein, the additional application server is similar to server 145 which includes Application/Dynamic content module 215.] The pool includes a first real application server having an application installed therein and communicating with a real data-source server to obtain application data from the data-source server. [Application servers 145, 150 and 155 of Figure 1 are in a cluster. Page 7 lines 11-25. Application/Dynamic content module 215 within application servers 145, 150 and 155. Page 7 lines 26- 42. Data source server 175 or 180. Page 5 lines 36-37. Page 8 lines 10-15. Figure 1 illustrates a network path including an optional router or firewall 165 and optional additional network 170 which interconnects application servers 145, 150 and 155 to data-source servers 175 and 180. Page 5 line 40 to Page 6 line 2.] The additional application server has the application installed therein. [Page 8 line 38 to Page 9

line 27. As indicated therein, the additional server is similar to server 145 which includes Application/Dynamic content module 215.] The real management server includes a CPU, a computer readable memory and a computer readable storage media. [CPU 231, RAM 233 and storage 234 added to Figure 2 by Amendment of 8/29/08 and corresponding addition to Specification on Page 5 line 26 by Amendment of 8/29/08.] The computer program product includes a computer readable storage media. [RAM 233 and storage 234 added to Figure 2 by Amendment of 8/29/08 and corresponding addition to Specification on Page 5 line 26 by Amendment of 8/29/08.] First program instructions receive performance data for the first real application server and performance data for the real data-source server. [Management server 140 of Figure 2. Page 8 lines 16-36. Analyser module 230 collects performance data. Steps 500, 505 and 510. Reporting module 205 of Figure 2 or Reporting module 210 of Figure 3. Step 355 of Figure 3. Steps 400, 405 and 410 of Figure 4. Page 7 line 42 to Page 8 line 5. Page 8 lines 16-26. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] Second program instructions, based on the performance data for the first real application server and the performance data for the real data-source server, determine that the first real application server is functional but has reached a predetermined upper level of utilization. [Analyser Module 230 of Figures 2 and 5. Page 8 lines 28-36. Steps 500, 510 and 515.] The performance data for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. [Page 7 line 42 to Page 8 line 5. Page 8 lines 16-26. Steps 500, 505 and 510. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] In response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization, the second program instructions identify the additional real application server as having the application but not currently allocated to the pool, [Resource Allocation module 235 in server 140. Page 8 line 38 to Page 9 line 15. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 21.] and the second program instructions select the real data-source server to provide application data to the additional real application server and automatically send connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server. [Resource Allocation

module 235 in server 140. Dynamic Configuration module 216. Page 8 line 38 to Page 9 line 27. Figure 3 illustrates "Connection Configuration" file 320. Page 8 lines 7-9. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 20.] The first and second program instructions are stored on the computer readable storage media. [RAM 233 and storage 234 added to Figure 2 by Amendment of 8/29/08 and corresponding addition to Specification on Page 5 line 26 by Amendment of 8/29/08. Page 4 line 19 - text added by Amendment of March 13, 2009 from original claims 1 and 20.]

VI. Grounds of Rejection to be Reviewed Upon Appeal

Claims 63-77 were rejected under 35 USC 103(a) based on US 6,859,830 to Ronneburg et al. and US Publication 2002/0032787 by Overton et al.

Claims 66, 71 and 76 were rejected under 35 USC 103(a) based on US 6,859,830 to Ronneburg et al., US Publication 2002/0032787 by Overton et al. and US Publication 2003/0126202 by Watt et al.

VII. Argument

A claim cannot be anticipated under 35 USC 102 unless each and every element as recited in the claim is found in a single prior art reference. Richardson v. Suzuki Motor Co., 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

A claim cannot be obvious under 35 USC 103 unless (a) there is a reason that a person of ordinary skill in the art would have combined the references, and (b) all the claim elements are taught or suggested by the prior art. See In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438, 1443 (Fed Cir. 1991) and KSR Int'l Co. v. Teleflex, Inc., No. 04-1350 (USSC 30 April 2007). Otherwise, there is not a prima facie case of obviousness.

Rejection of Claims 63 and 65-66, 68 and 70-71, 73 and 75-76 under 35 USC 103(a) Based on US 6,859,830 to Ronnenburg et al. and US Publication 2002/0032787 by Overton et al.

According to claim 63, the real management server, based on the performance data for the first real application server and the real data-source server, automatically determines that the first real application server is functional but has reached a predetermined upper level of utilization. The performance data for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. The real management server selects the real data-source server to provide application data for an additional real application server and sends connection settings for the real data-source server to the additional real application server. Ronnenburg et al. and Overton et al. in combination do not teach or suggest these features of claim 63.

Ronnenburg et al. disclose a server pool where each server in the pool is responsible for monitoring the up or down status of two "buddy" servers, based on responses to pings sent to the buddy servers. If a "buddy" server does not respond to the ping, it is removed from the server pool. A new server can be added to the pool by announcing itself to a server table. After being added to the pool, it is assigned buddies whose status it checks. Ronnenburg et al. state,

> "A virtual ring structure. In the virtual ring structure, each server is only required to monitor the status of two other servers in the serve pool. Thus, a server need only transmit ping signals to two other servers (its buddies) in the server pool at any given time. Because each server maintains the status of only two other servers at any given time, the size of the server pool is not limited by the ability of each server to send and process ping signals." Ronnenburg et al. Column 2 lines 8-17.

> "The server table maintains a list of each "live" server and the buddy servers assigned to that server. Down servers are removed from the server table, and thus, the server pool, by use of the server table within the SQL server. When a server determines that one of its buddies is down, the report to the SQL server results in a buddy reassignment. The

buddies of the down server are made buddies of one another and the virtual server ring is established. The SQL server then knows not to route any client to the down server." Ronnenburg et al. Column 2 lines 25-33.

"When a server is to be added to the server pool, another buddy reassignment is required. In such a case, a server and its buddy will be reassigned the added server as a buddy. Thus, the added server will have the original server and the buddy server as its buddies." Ronnenburg et al. Column 2 lines 34-39.

"FIG. 3 is a flowchart illustrating a method 300 for adding a new server to the virtual server ring in accordance with an embodiment of the present invention. The method 300 begins at start step 305 and proceeds to step 310 when a new server announces itself to the server table 135. Typically, on server start-up, the new server announces itself to the server table using an ADO ("Active Data Objects") call to ServerAnnounce, passing the new server's IP address as an argument. ADO is a programming interface that is designed by the Microsoft Corporation of Redmond, Washington as a standard for data access, which allows a client to interact with a SQL server, such as the Web server 110. In the preferred embodiment of the present invention, the ServerAnnounce stored procedure is stored on the Web server 110. The method proceeds to step 315. At step 320, the buddies of the new server are returned to the new server by the return mechanism of the call to the ServerAnnounce procedure. At this point, the new server becomes responsible for looking after these buddies." Ronnenburg et al. Column 5 lines 15-38.

While Ronnenburg et al. determine whether each server is functional based on the pings, Ronnenburg et al. do not determine the performance of the first real application server based on the performance data for the first real application server and performance data for the real data-source server. (This determines whether slow performance of the first real application server is due to CPU constraint of the first real application server or delays in the real data-source server furnishing data to the first real application server needed to comply with the client request to the real application server.) Ronnenburg et al. do not even mention real data-source servers

that supply data to the real application servers; rather, Ronnenburg et al. are only concerned with one "level" of server. Also, Ronnenburg et al. do not disclose or suggest that a real management server selects a real data-source server for a real application server added to the pool, nor that the real management server sends connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server. Ronnenburg et al. fail to disclose these key functions of a management server. Overton et al. do not fill these gaps of Ronnenburg et al.

Overton et al. disclose that a client makes a request to a system to identify the location of a data repository containing data of the client, and the system responds with the identity of the data repository so the client can contact the data repository directly,

> "In a networked environment where there are a large number of data repositories and any particular entity does not have data in all of the data repositories, a mechanism is needed that would permit queries to be directed only at data repositories with relevant information. It would also be beneficial to permit membership in the set of data repositories itself to be highly dynamic. Such a system would support on-the-fly addition and removal of data repositories from the topology of a distributed database seamlessly and without the need to reprogram the database." Overton et al. Paragraph [0006]

> "A system for managing location information and providing location information to data location queries comprises a transfer protocol configured to manipulate an identifier, and at least one location associated with the identifier, wherein the identifier uniquely specifies an entity and wherein each data location specifies a location of data in a network pertaining to the entity. The system also includes a location server containing location information corresponding to at least on entity that is formatted according to the transfer protocol, wherein the location of data relates to an application server in the network. The system further includes programming logic stored on the location server that is responsive to a location query identifying a desired entity to return a location message. The location

message includes one or more locations associated with the desired entity." Overton et al. Paragraph [0008].

"The method includes receiving a location query from a client requesting the location of data relevant to an entity identified in the query. The queried location server sends a location response message to the client if the queried location server contains information relevant to the entity identified in the query. The location server sends a redirect message to the client if it does not contain location information relevant to the entity identified in the query, where the redirect message comprising a list of location servers containing information relevant to the entity identified in the query." Overton et al. Paragraph [0009].

"When a client queries the network distributed tracking protocol ("NDTP") server for information pertaining to an entity, the NDTP server preferably returns a list of all locations for the specified entity. The client then can directly access the various locations, relieving the NDTP server of any further involvement in the transaction and allowing the NDTP server to handle more queries." Overton et al. Paragraph [0040]

Thus, Overton et al. are concerned with identifying to a client/requestor the location of a data repository containing data of the client, so the client can contact the data repository directly to access the data. But this is unrelated to identifying a backend data-source server that provides data to an application server added to a pool. In contrast to claim 63, Overton et al. do not disclose or even suggest that a real management server determines that a first real application server has reached a predetermined upper level of utilization, based on the performance data for the first real application server and performance data for the real data-source server, where the performance for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. (Overton et al. do not even determine performance data for a functional computer.) Overton et al. do not disclose or suggest that a real management server selects a real data-source server for a real application server added to the pool, nor that the real management server sends

connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server. Therefore, the rejection under 35 USC 103(a) based on Ronnenburg et al. and Overton et al. should be reversed.

Independent claims 68 and 73 distinguish over Ronnenburg et al. and Overton et al. for the same reasons that claim 63 distinguishes thereover.

Claims 65, 70 and 75 depend on claims 63, 68 and 73, respectively, and therefore, distinguish over Ronnenburg et al. and Overton et al. for the same reasons that claims 63, 68 and 73 distinguish thereover.

Claims 66, 71 and 76 depend on claims 63, 68 and 73, respectively, and therefore, distinguish over Ronnenburg et al. and Overton et al. for the same reasons that claims 63, 68 and 73 distinguish thereover.

Rejection of Claims 64, 69 and 74 under 35 USC 103(a)
Based on US 6,859,830 to Ronnenburg et al. and US Publication 2002/0032787 by Overton et al.

Claims 64, 69 and 74 depend on claims 63, 68 and 73, respectively, and therefore, distinguish over Ronnenburg et al. and Overton et al. for the same reasons that claims 63, 68 and 73 distinguish thereover.

Claim 64 further distinguishes over Ronnenburg et al. and Overton et al. by reciting that in response to the first real application server reaching a predetermined upper level of utilization, the real management server automatically sends to the additional real application server port settings for the real data-source server to communicate with the real data-source server to obtain application data from the real data-source server. Neither Ronnenburg et al. nor Overton et al. teach or suggest this feature of claim 64.

Claims 69 and 74 further distinguish over Ronnenburg et al. and Overton et al. for the same reasons that claim 64 further distinguishes thereover.

Rejection of Claims 67, 72 and 77 under 35 USC 103(a)
Based on US 6,859,830 to Ronnenburg et al. and US Publication 2002/0032787 by Overton et al.

Claims 67, 72 and 77 depend on claims 63, 68 and 73, respectively, and therefore, distinguish over Ronnenburg et al. and Overton et al. for the same reasons that claims 63, 68 and 73 distinguish thereover.

Claim 67 further distinguishes over Ronnenburg et al. and Overton et al. by reciting that in response to the first real application server reaching a predetermined upper level of utilization, the real management server automatically sends to the additional real application server a description of an installation path for the real data-source server to support communication with the additional real application server. Neither Ronnenburg et al. nor Overton et al. teach or suggest this feature of claim 67.

Claims 72 and 77 further distinguish over Ronneburg et al. and Overton et al. for the same reasons that claim 67 further distinguishes thereover.

Rejection of Claims 66, 71 and 76 under 35 USC 103(a)
Based on US 6,859,830 to Ronneburg et al., US Publication 2002/0032787 by Overton et al. and US Publication 2003/0126202 by Watt et al.

Claims 66, 71 and 76 depend on claims 63, 68 and 73, respectively. Claim 66 recites that a multiplicity of copies of the application are installed in a respective multiplicity of the real application servers in the pool. The Examiner cited Watt et al. as teaching this feature.

Watt et al. disclose a load manager which allocates and de-allocates servers in a pool, but does not disclose a data-source server which supplies data to an application server in the pool,

"Each server manager 208a is responsible for rebooting or powering on and off specified servers as directed by load manager 206." Watt et al. Paragraph [0047]

"Server monitor 204 continuously monitors the health, load and response time for all servers within data center 200. It detects and reports server failures to load manager 206. Server monitor 204 also calculates average server load and response times for each configured server pool 212, reporting under and over-load conditions to load manager 206." Watt et al. Paragraph [0048]

"Load manager 206 is responsible for allocating servers and images 217 to meet the requirements of data center 200." ... Load manager 206 powers servers on or off as needed via the server manager 208a." Watt et al. Paragraph [0049]

"Infrastructure controller 202 is responsible for configuring the network infrastructure 214 and 216 surrounding a server to provide secure, limited access to those resources required by the server and its applications. This includes configuring network switches and virtual LANs (VLANs) 216. The tasks of infrastructure controller 202 also include configuring load balancers 216 to add/remove servers from the affected server pools 212, configuring all switch ports connected to the server to ensure that the server and its applications have access to the network resources they need, and to prevent them from accessing any restricted resources that they are not authorized to access." Watt et al. Paragraph [0051]

"In an embodiment, software to be executed is installed only once, regardless of how many servers will eventually execute the software. This master installation is called a "snapshot" (FIG. 3 shows snapshots 302a-n representing several different installed software applications for execution on various servers 308a-n within the data center 200)." Watt et al. Paragraph [0061]

"Server images 217 are generated in an automated manner from a server class 304. A working server image 217 is called an "instance" 306 of the server class 304. When a server 308 boots, it mounts an instance 306." Watt et al. Paragraph [0062]

"DSAP system 102 provisions a server 308 by generating a fully configured, bootable instance 306 of the appropriate server class 304, complete with network address assignments, VLAN configuration, load balancing configuration, etc. Provisioning n instances 306 of a server class 304 provides DSAP system 102 with the capacity to run n servers 308 of the specified class 304, provided that sufficient server resources are available. To execute those instances, however, the required n number of servers 308 must first be allocated by assigning them an instance 306." Watt et al. Paragraph [0094]

"A local instance 306 is an independent instance that is physically stored on the local storage attached to a server 308. Because the local instance physically resides with the server, it can only be run by that server and cannot be allocated elsewhere." Watt et al. Paragraph [0096]

Thus, Watt et al. disclose a load manager which allocates and de-allocates servers in a pool, but does not disclose monitoring performance of a data-source server to determine impact on performance of an application server. Watt et al. do not disclose or suggest: The performance data for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. The real management server selects the real data-source server to provide application data for an additional real application server and sends connection settings for the real data-source server to the additional real application server. Therefore, Watt et al. do not fill the foregoing gaps of Ronnenburg et al. and Overton et al. relative to independent claims 63, 68 and 73.

Based on the foregoing, Appellants request reversal of all rejections made by the Examiner.

Respectfully submitted,

Dated: April 19, 2010
Phone: 607-429-4368
Fax: 607-429-4119

/Arthur J. Samodovitz/
Arthur J. Samodovitz
Reg. No. 31,297

VIII. Claims Appendix

63. A method for allocating an additional real application server to an existing pool of real application servers, the pool including a first real application server having an application installed therein and communicating with a real data-source server to obtain application data from the data-source server, the additional application server having the application installed therein, the method comprising the steps of:

a real management server for the pool receiving performance data for the first real application server and performance data for the real data-source server;

the real management server, based on the performance data for the first real application server and the performance data for the real data-source server, automatically determining that the first real application server is functional but has reached a predetermined upper level of utilization, the performance data for the real data-source server indicating an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool, and in response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization,

the real management server automatically identifying the additional real application server as having the application but not currently allocated to the pool, and

the real management server automatically selecting the real data-source server to provide application data to the additional real application server and automatically sending connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server.

64. The method set forth in claim 63 wherein:

in response to the step of the real management server automatically determining that the first real application server is functional but has reached a predetermined upper level of utilization, further comprising the step of the real management server automatically sending to the additional real application server, port settings for the real data-source server to communicate with the real data-source server to obtain application data from the real data-source server.

65. The method set forth in claim 63 further comprising the subsequent steps of:

based on subsequent performance data of the first real application server, the real management server determining that the first real application server is functional but under utilized such that the first real application server is no longer needed in the pool, and in response, the real management server automatically de-allocating the first real application server from the pool.

66. The method set forth in claim 63 wherein a multiplicity of copies of the application are installed in a respective multiplicity of the real application servers in the pool.

67. The method set forth in claim 63 wherein:

in response to the step of the real management server automatically determining that the first real application server is functional but has reached a predetermined upper level of utilization, further comprising the step of the real management server automatically sending to the additional real application server a description of an installation path for the real data-source server to support communication with the additional real application server.

68. A real management server for allocating an additional real application server to an existing pool of real application servers, the pool including a first real application server having an application installed therein and communicating with a real data-source server to obtain application data from the data-source server, the additional application server having the application installed therein, the real management server comprising:

a CPU, a computer readable memory and a computer readable storage media;

first program instructions to receive performance data for the first real application server and performance data for the real data-source server;

second program instructions, based on the performance data for the first real application server and the performance data for the real data-source server, to determine that the first real application server is functional but has reached a predetermined upper level of utilization, the performance data for the real data-source server indicating an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool, and in response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization,

the second program instructions identify the additional real application server as having the application but not currently allocated to the pool, and

the second program instructions select the real data-source server to provide application data to the additional real application server and automatically send connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server; and wherein

the first and second program instructions are stored on the computer readable storage media for execution by the CPU via the computer readable memory.

69. The real management server set forth in claim 68 wherein:

in response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization, the second program instructions automatically

send to the additional real application server, port settings for the real data-source server to communicate with the real data-source server to obtain application data from the real data-source server.

70. The real management server set forth in claim 68 wherein:

the second program instructions, based on subsequent performance data of the first real application server, determine that the first real application server is functional but under utilized such that the first real application server is no longer needed in the pool, and in response, the second program instructions automatically de-allocate the first real application server from the pool.

71. The real management server set forth in claim 68 wherein a multiplicity of copies of the application are installed in a respective multiplicity of the real application servers in the pool.

72. The real management server set forth in claim 68 wherein:

the second program instructions, responsive to the determination that the first real application server is functional but has reached a predetermined upper level of utilization, automatically send to the additional real application server a description of an installation path for the real data-source server to support communication with the additional real application server.

73. A computer program product for execution in a real management server for allocating an additional real application server to an existing pool of real application servers, the pool including a first real application server having an application installed therein and communicating with a real data-source server to obtain application data from the data-source server, the additional application server having the application installed therein, the computer program product comprising:

a computer readable storage media;

first program instructions to receive performance data for the first real application server and performance data for the real data-source server;

second program instructions, based on the performance data for the first real application server and the performance data for the real data-source server, to determine that the first real application server is functional but has reached a predetermined upper level of utilization, the performance data for the real data-source server indicating an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool, and in response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization,

the second program instructions identify the additional real application server as having the application but not currently allocated to the pool, and

the second program instructions select the real data-source server to provide application data to the additional real application server and automatically send connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server; and wherein

the first and second program instructions are stored on the computer readable storage media.

74. The computer program product set forth in claim 73 wherein:

in response to the determination that the first real application server is functional but has reached a predetermined upper level of utilization, the second program instructions automatically send to the additional real application server, port settings for the real data-source server to communicate with the real data-source server to obtain application data from the real data-source server.

75. The computer program product set forth in claim 73 wherein:

the second program instructions, based on subsequent performance data of the first real application server, determine that the first real application server is functional but under utilized such that the first real application server is no longer needed in the pool, and in response, the second program instructions automatically de-allocate the first real application server from the pool.

76. The computer program product set forth in claim 73 wherein a multiplicity of copies of the application are installed in a respective multiplicity of the real application servers in the pool.

77. The computer program product set forth in claim 73 wherein:

the second program instructions, responsive to the determination that the first real application server is functional but has reached a predetermined upper level of utilization, automatically send to the additional real application server a description of an installation path for the real data-source server to support communication with the additional real application server.

IX. Evidence Appendix

There is no evidence entered or relied upon in this Appeal.

X. Related Proceedings Appendix

There are no related Appeals or Interferences, and therefore, no copies of such decisions.